A Newton-Krylov Multigrid Method for the Incompressible Navier-Stokes Equations*

Michael Pernice
Center for High Performance Computing
University of Utah
pernice@chpc.utah.edu

Michael D. Tocci
Department of Mathematical Sciences
Worcester Polytechnic Institute
mdtocci@wpi.edu

and

Center for the Simulation of Accidental Fires & Explosions
University of Utah

*Work supported the University of Utah Center for the Simulation of Accidental Fires & Explosions, funded by the Department of Energy, Lawrence Livermore National Laboratory, under subcontract B341493, and the University of Utah Center for High Performance Computing. The first author received additional support through the Department of Energy Office of Energy Research under grant DE–FG03–96ER82268,

1999 ASCI Tri-lab Workshop on Solvers

Overview

- Inexact Newton method
- Linear multigrid preconditioner
- Pressure-correction smoother
- Numerical examples
- Implementation extensions

Notation

The steady-state incompressible Navier-Stokes equations:

$$\begin{aligned} (uu)_x + (uv)_y - \tfrac{1}{Re}\Delta u + p_x &= b_1 \\ (uv)_x + (vv)_y - \tfrac{1}{Re}\Delta v + p_y &= b_2 \\ u_x + v_y &= 0. \end{aligned}$$

Second-order centered discretization on a staggered grid produces a set of nonlinear equations

$$\begin{aligned} F(u,v,p) &= \begin{pmatrix} Q_1[\mathbf{u}] & 0 & \mathcal{G}_x^h \\ 0 & Q_2[\mathbf{u}] & \mathcal{G}_y^h \\ \mathcal{D}_x^h & \mathcal{D}_y^h & 0 \end{pmatrix} \begin{pmatrix} u \\ v \\ p \end{pmatrix} - \begin{pmatrix} b_1 \\ b_2 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{Q}[\mathbf{u}] & \nabla^h \\ \nabla^h\cdot & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} - \begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix}. \end{aligned}$$

Globalized Inexact Newton Method

ALGORITHM: INEXACT NEWTON BACKTRACKING (INB) [EW96]

Let x_0, $\epsilon > 0$, $\eta_{max} \in [0,1)$, $t \in (0,1)$ and $0 < \theta_{min} < \theta_{max} < 1$ be given.
Set $k = 0$.
While $\|F(x_k)\| > \epsilon$ do:
  Choose **initial** $\eta_k \in [0, \eta_{max}]$ and s_k such that
    $\|F(x_k) + F'(x_k)s_k\| \leq \eta_k \|F(x_k)\|$.
  While $\|F(x_k + s_k)\| > [1 - t(1 - \eta_k)]\|F(x_k)\|$ do:
    Choose $\theta \in [\theta_{min}, \theta_{max}]$.
    Update $s_k \leftarrow \theta s_k$ and $\eta_k \leftarrow 1 - \theta(1 - \eta_k)$.
  Set $x_{k+1} = x_k + s_k$.
  $k = k + 1$.

Choosing the Forcing Terms

Several options for selecting $\{\eta_k\}$ are available. This study uses

$$\eta_k = \min\left\{ \eta_{max}, \frac{\left| \, \|F(x_k)\| - \|F(x_{k-1}) + F'(x_{k-1})s_{k-1}\| \, \right|}{\|F(x_{k-1})\|} \right\}.$$

To prevent η_k from getting too small too soon, this is safeguarded with

$$\eta_k = \min\left\{ \eta_{max}, \max\{\eta_k, \eta_{k-1}^{(1+\sqrt{5})/2}\} \right\} \text{ if } \eta_k \geq threshold.$$

It can be shown that superlinear convergence of the inexact Newton method is obtained with this choice of $\{\eta_k\}$ [EW96].

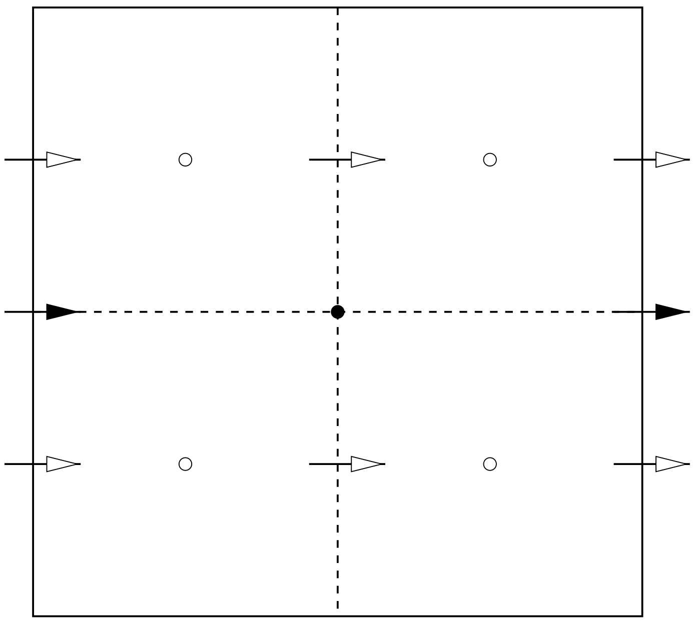
Linear Multigrid Preconditioner

Problem statement: solve a system of linear equations $Lx = f$.

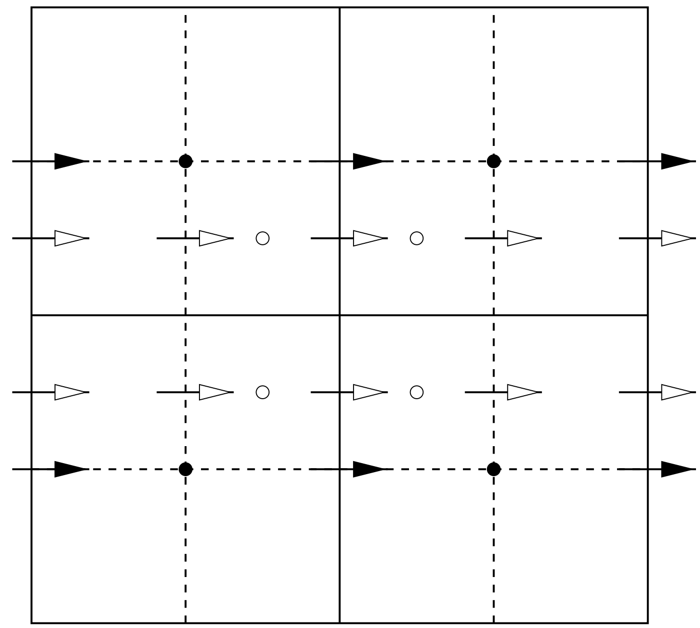
<u>ALGORITHM</u>: LINEAR MULTIGRID V-CYCLE

PROCEDURE MG-V(h, L^h, x^h, f^h)
If $h = h_c$ then:
  Solve $L^h x^h = f^h$.
else
  Presmooth $x^h \longleftarrow x^h + B(f^h - L^h x^h)$ ν_1 times.
  Set $x^{2h} = 0$.
  Restrict $f^{2h} = I_h^{2h}(f^h - L^h x^h)$.
  MG-V($2h$, L^{2h}, x^{2h}, f^{2h}).
  Correct $x^h = x^h + I_{2h}^h x^{2h}$.
  Postsmooth $x^h \longleftarrow x^h + B(f^h - L^h x^h)$ ν_2 times.

Intergrid Transfers on a Staggered Grid



Restriction



Prolongation

Pressure-correction Smoother

The SIMPLE method starts by solving

$$\mathbf{Q}[\mathbf{u}^{(n)}]\mathbf{u}^{(n+\frac{1}{2})} = \mathbf{b} - \nabla^h p^{(n)}.$$

Next, find a correction δp to the pressure, and also use its gradient to correct $\mathbf{u}^{(n+\frac{1}{2})}$.

$$\begin{aligned}
\mathbf{u}^{(n+\frac{1}{2})} &\approx \mathbf{D}^{-1}\mathbf{Q}\mathbf{u}^{(n+\frac{1}{2})} &= \mathbf{D}^{-1}(\mathbf{b} - \nabla^h p^{(n)}) \\
\mathbf{u}^{(n+1)} &\approx \mathbf{D}^{-1}\mathbf{Q}\mathbf{u}^{(n+1)} &= \mathbf{D}^{-1}(\mathbf{b} - \nabla^h p^{(n+1)})
\end{aligned}$$

$$\delta\mathbf{u} \equiv \mathbf{u}^{(n+1)} - \mathbf{u}^{(n+\frac{1}{2})} = -\mathbf{D}^{-1}\nabla^h \delta p$$

The Pressure Correction Step in SIMPLE

Apply $\nabla^h\cdot$ to this and require $\nabla^h \cdot \mathbf{u}^{(n+1)} = 0$ to obtain

$$S\delta p = -\nabla^h \cdot \mathbf{u}^{(n+\frac{1}{2})}$$

where

$$S = -\nabla^h \cdot \mathbf{D}^{-1}\nabla^h.$$

Once the pressure update and the velocity corrections are obtained, the pressure and velocity fields are updated.

Practical implementations usually have to damp these corrections to stabilize the algorithm.

SIMPLE Uses a Projection

Let $\mathcal{P} = I + \mathbf{D}^{-1}\nabla^h S^{-1}\nabla^h\cdot$. Then

$$\mathbf{u}^{(n+1)} = \mathcal{P}\mathbf{u}^{(n+\frac{1}{2})}$$

and

$$\begin{aligned}
\mathcal{P}^2 &= I + 2\mathbf{D}^{-1}\nabla^h S^{-1}\nabla^h \cdot + \left(\mathbf{D}^{-1}\nabla^h S^{-1}\nabla^h\cdot\right)\left(\mathbf{D}^{-1}\nabla^h S^{-1}\nabla^h\cdot\right) \\
&= I + \mathbf{D}^{-1}\nabla^h S^{-1}\nabla^h \cdot \\
&= \mathcal{P}
\end{aligned}$$

so $\mathcal{P}$ is a *projection*, but it is *not* an orthogonal projection w.r.t the standard inner product.

Newton-Krylov-Multigrid Methods

In a linear multigrid preconditioner with SIMPLE smoothing, compute $Q[\mathbf{u}^k]$ after each Newton step and use it in the multigrid preconditioner.[1]

Alternatively, a lower-order discretization $Q_{FOU}[\mathbf{u}^k]$ can be computed in the setup phase of the preconditioner. Storage for $Q[\mathbf{u}^k]$ can be re-used in the preconditioner, and $Q[\mathbf{u}^k]$ can be restored after the preconditioner is applied.

Thus,

- storage overhead and initialization of the multigrid preconditioner is minimal; and
- no explicit representation of the Jacobian is used.

[1] Thanks to D. Knoll for pointing out this would work.

Example: Bouyancy-driven Natural Convection on $\Omega = [0,1]^2$

$$\begin{aligned}
(uu)_x + (uv)_y + p_x - \tfrac{1}{Re}\Delta u &= 0 \\
(uv)_x + (vv)_y + p_y - \tfrac{1}{Re}\Delta v - \tfrac{Ra}{Re^2 Pr} T &= 0 \\
u_x + v_y &= 0 \\
(uT)_x + (vT)_y - \tfrac{1}{RePr}\Delta T &= 0
\end{aligned}$$

$$\begin{array}{ll}
u = v = 0 & \text{on } \partial\Omega \\
T(0,y) = 0, \qquad T(1,y) = 1 & y \in [0,1] \\
T_y(x,0) = T_y(x,1) = 0 & x \in [0,1]
\end{array}$$

Performance Statistics for $\mathrm{Ra} = 100,000$

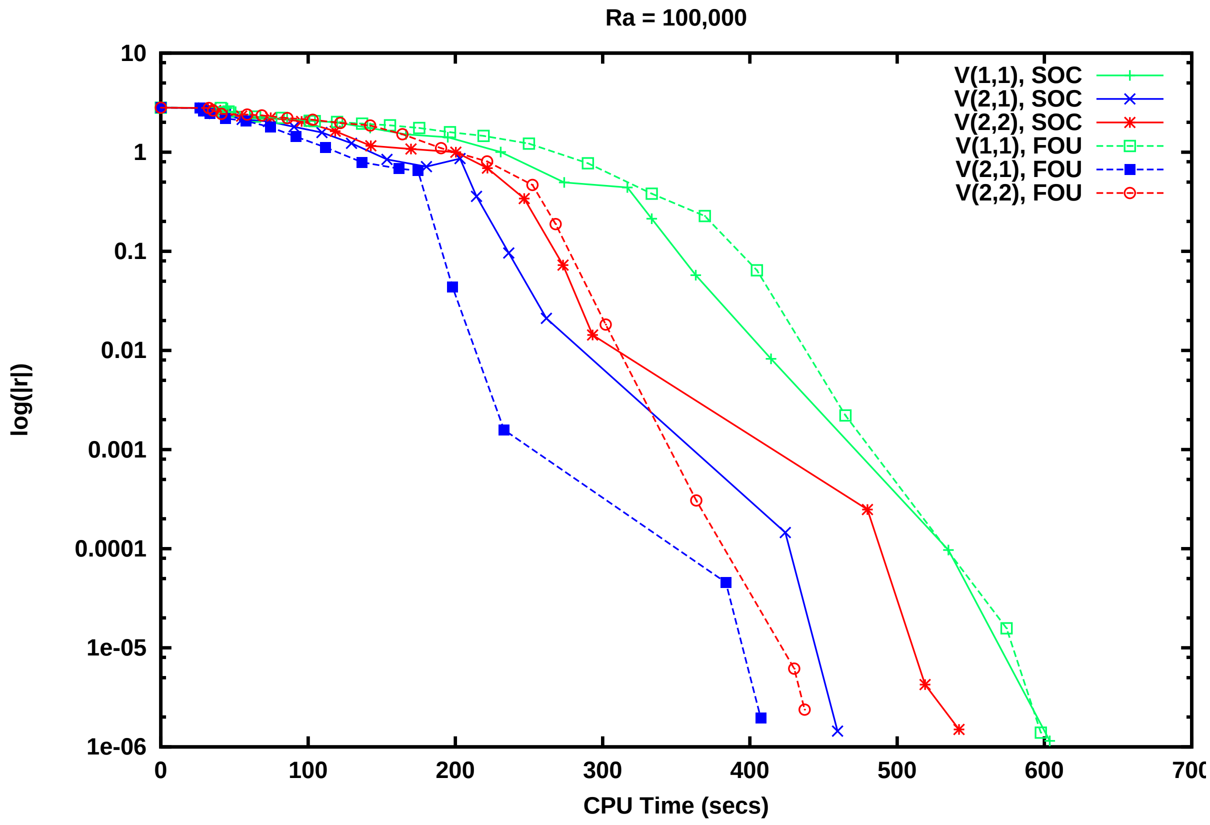
Precondition with Q

	SG-1	SG-2	SG-4	V(1,1)	V(2,1)	V(2,2)	V(4,2)	V(4,4)
NLI	3142	1165	660	552	327	318	250	303
NNI	30	31	33	18	17	17	19	16
NBT	8	8	6	3	3	3	3	2
T	2255	1009	847	604	460	542	564	879

Precondition with Q_{FOU}

	SG-1	SG-2	SG-4	V(1,1)	V(2,1)	V(2,2)	V(4,2)	V(4,4)
NLI	3341	1182	665	593	304	277	232	281
NNI	32	33	32	21	15	18	18	18
NBT	5	8	6	4	2	3	3	3
T	2409	996	831	598	407	437	507	775

Convergence Histories for $\mathrm{Ra} = 100,000$



Performance Statistics for $\mathrm{Ra} = 1,000,000$

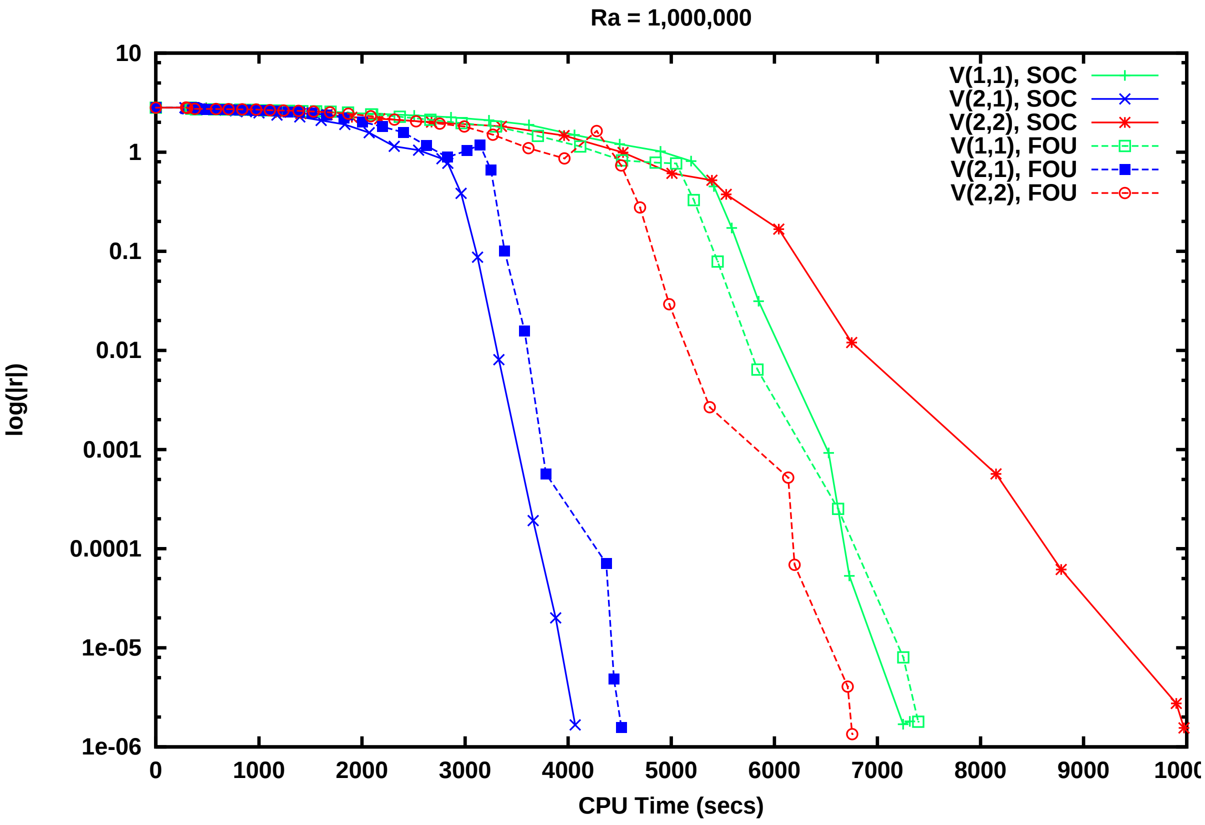
Precondition with Q

	V(1,1)	V(2,1)	V(2,2)	V(4,2)	V(4,4)
NLI	1355	695	1335	618	842
NNI	27	26	27	27	27
NBT	6	5	5	5	5
T	7314	4068	9974	5482	10270

Precondition with Q_{FOU}

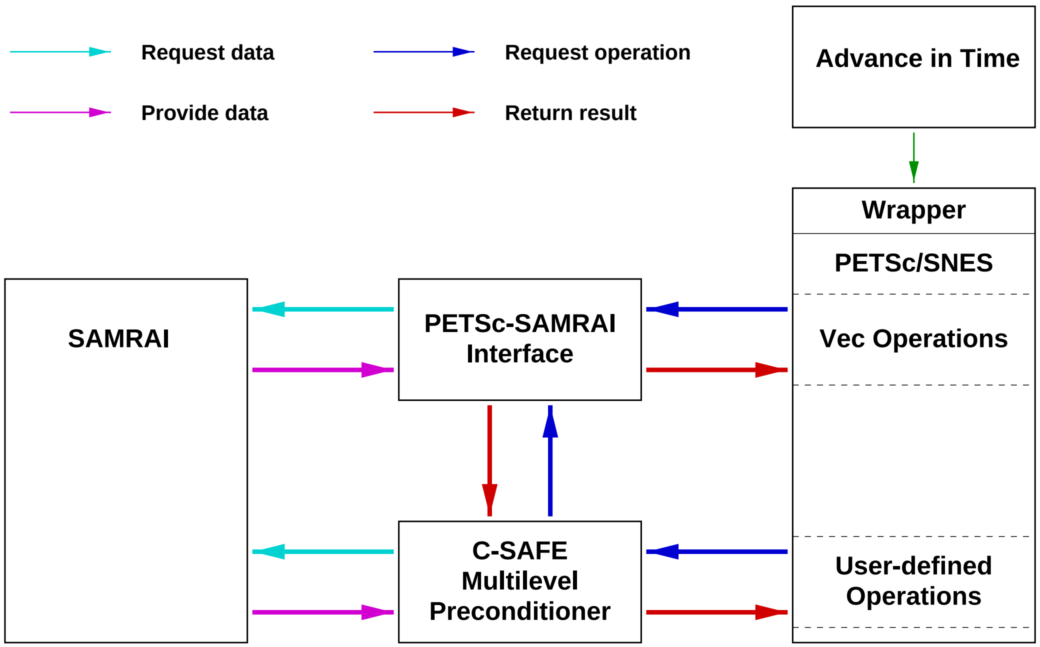
	V(1,1)	V(2,1)	V(2,2)	V(4,2)	V(4,4)
NLI	1738	887	1055	809	1059
NNI	30	32	30	28	26
NBT	6	6	6	5	4
T	7395	4516	6756	6750	11250

Convergence Histories for $\mathrm{Ra} = 1,000,000$



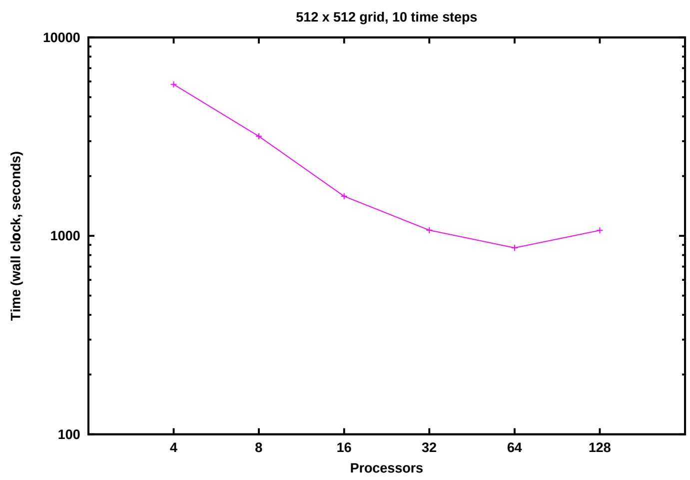
Extensions: Interfaces

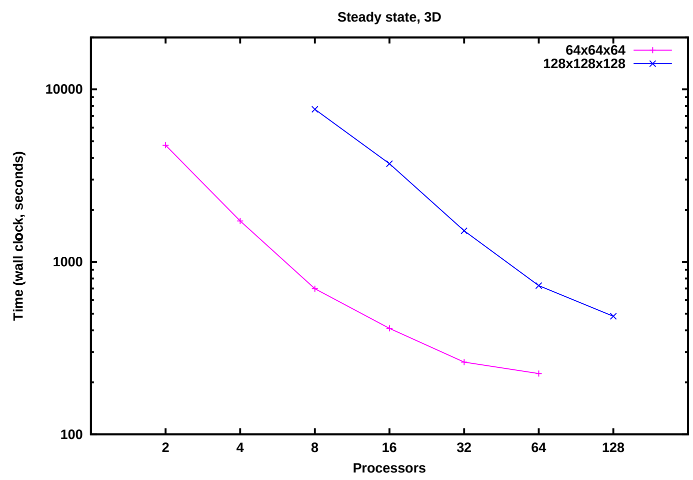
We are currently extending these ideas to parallel computation of unsteady flow on block structured adaptive grids. Implementation is based on the `SAMRAI` framework and an interface between `SAMRAI` and `PETSc`.



Extensions: Parallel Solution of Navier-Stokes Equations

Migration of these methods to the `SAMRAI` framework required some minor reorganization and creation of some additional `C++` infrastructure. These efforts led first to an unsteady solver, and subsequently to a parallel version that was also easily extended to treat 3D problems.





Extensions: Sensitivity Analysis

Objective: solve

$$F(t, y, y', p) = 0$$

where p is a vector of parameters.

Maly-Petzold (1996) algorithm: set

$$\begin{aligned} G_0 = F(t, y, y', p) &= 0 \\ G_i = \frac{\partial F}{\partial y} s_i + \frac{\partial F}{\partial y'} s_i' + \frac{\partial F}{\partial p_i} &= 0, \; i = 1, \ldots, m \end{aligned}$$

where $s = \left(\frac{\partial y}{\partial p_1} \ldots \frac{\partial y}{\partial p_m} \right)^T$ is a vector of sensitivities.

Strategy: estimate G_i with finite differences, and solve for y and s simultaneously in time.

Linear Structure

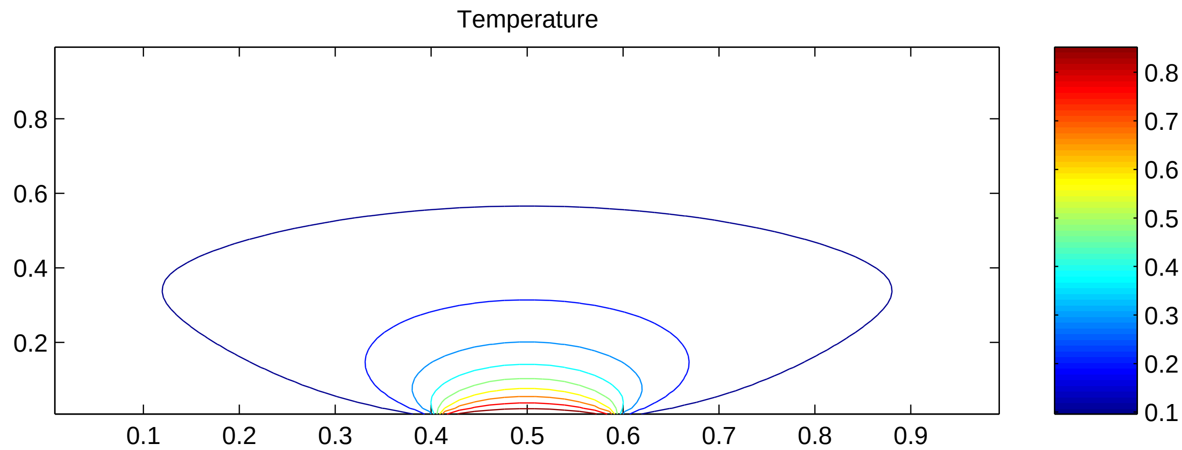
Let J^* be the Jacobian of the complete system, then solve

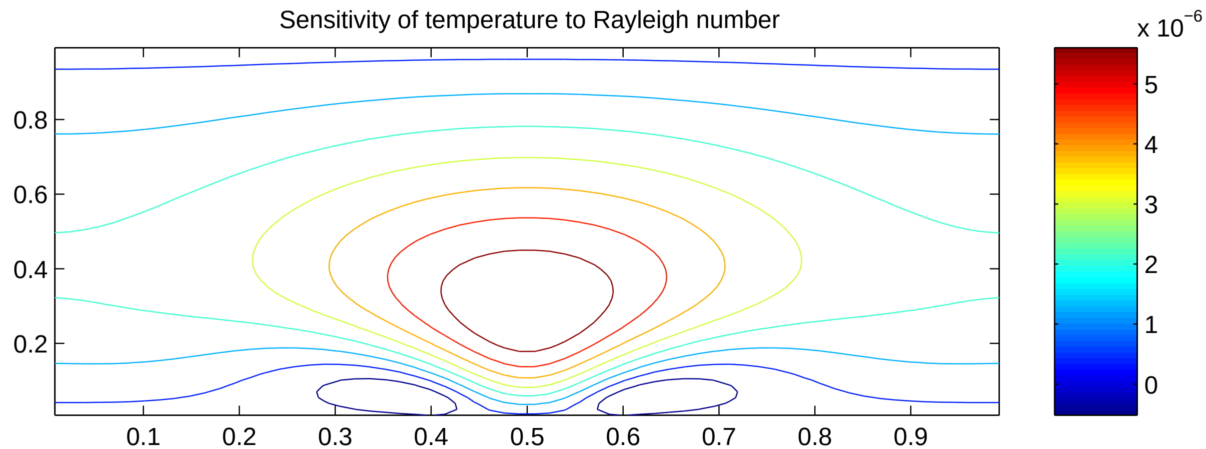
$$\begin{aligned} J^*\Delta &= -G \\ y^{k+1} &= y^k + \Delta_0 \\ s_i^{k+1} &= s_i^k + \Delta_i, \; i = 1, \ldots, m \end{aligned}$$

for a Newton-like iteration

- Advantage – uses the full Jacobian
  - ▷ approximate its action with finite differences
- Disadvantage – preconditioning can be difficult since J^* is complicated
  - ▷ use a block diagonal preconditioner with the MG-SIMPLE preconditioner in each block

Sample Results





Conclusions and Future Work

Multigrid methods are promising preconditioners for inexact Newton methods:

- highly effective;
- low startup costs;
- low storage overhead;
- can mix discretizations of different order;
- can reduce the storage overhead of an inexact Newton method.

Further work to be done:

- extension to SAMR methods through multilevel preconditioning;
- tuning parallel performance;
- further applications in sensitivity analysis and more realistic problems;
- improved, simplified user interfaces.